

Using the Forest to See the Trees: Context-based Object Recognition

Bill Freeman

Joint work with Antonio Torralba and Kevin Murphy

Computer Science and Artificial Intelligence

Laboratory

MIT

A computer vision goal

- Recognize many different objects under many viewing conditions in unconstrained settings.
- There has been progress on restricted cases:
 - one object and one pose (frontal view faces)
 - Isolated objects on uniform backgrounds.
- But the general problem is difficult and unsolved.

VISIT...

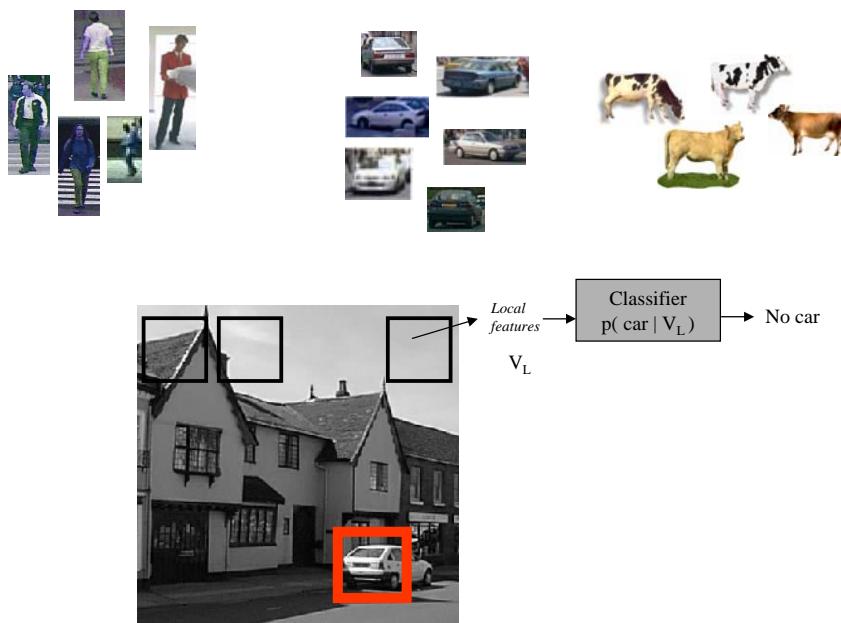


How we hope to make progress on this hard problem

- Various technical improvements
- Exploit scene context:
 - “if this is a forest, these must be trees”.

Local (bottom-up) approach to object detection

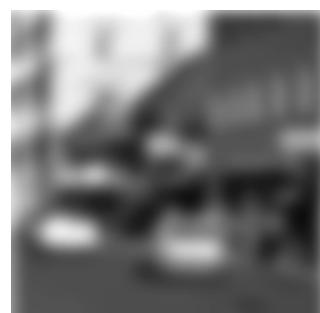
Classify image patches/features at each location and scale



Problem 1: Local features can be ambiguous



Solution 1: Context can
disambiguate local features



Effect of context on object detection



car



pedestrian

Identical local image features!

Images by Antonio Torralba

Even high-resolution images can
be locally ambiguous





Object in context



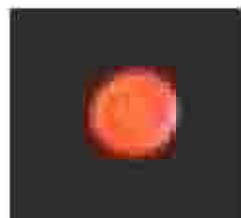
(Courtesy of Fredo Durand and William Freeman. Used with permission.)

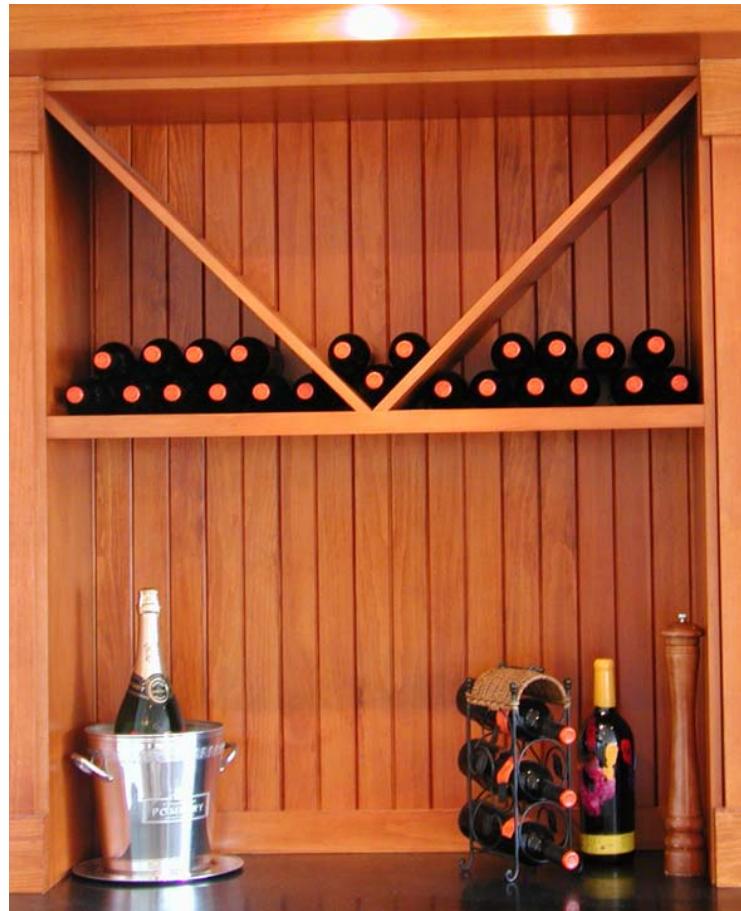
Isolated object



Object in context



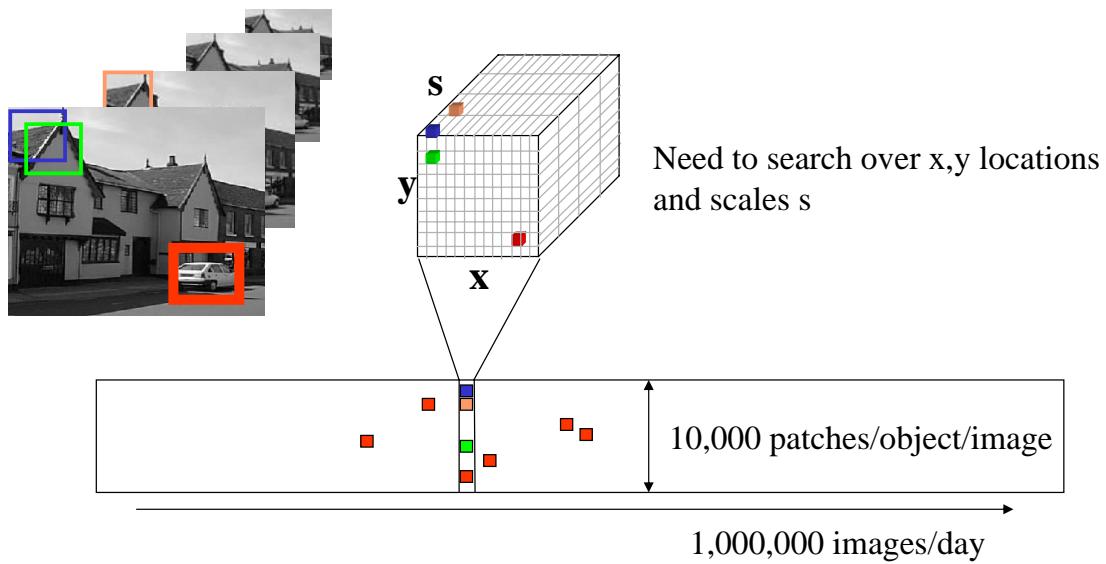




Problem 2: search space is **HUGE**

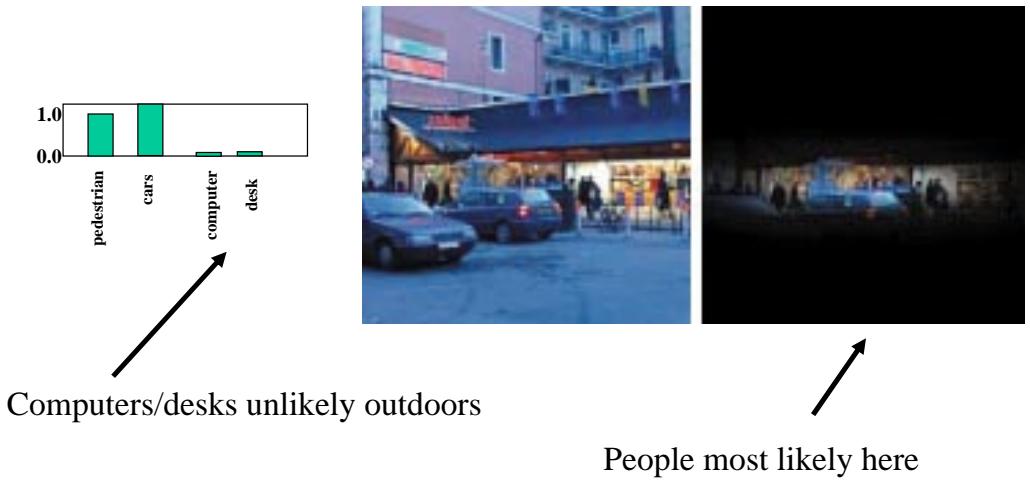
“Like finding needles in a haystack”

- Slow (many patches to examine)
- Error prone (classifier must have very low false positive rate)



Plus, we want to do this for ~ 1000 objects

Solution 2: context can provide a prior on what to look for, and where to look for it



Torralba, IJCV 2003

Talk outline

- Context-based vision
- Feature-based object detection
- Graphical model to combine both sources

Talk outline

- Context-based vision
- Feature-based object detection
- Graphical model to combine both sources

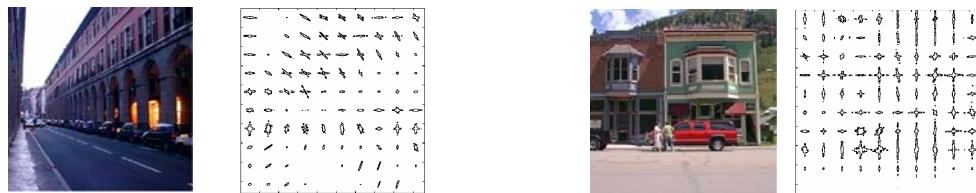
Context-based vision

- Measure overall scene context or “gist”
- Use that scene context for:
 - Location identification
 - Location categorization
 - Top-down info for object recognition
- Combine with bottom-up object detection
- Future focus: training set acquisition.

The “Visual Gist” System

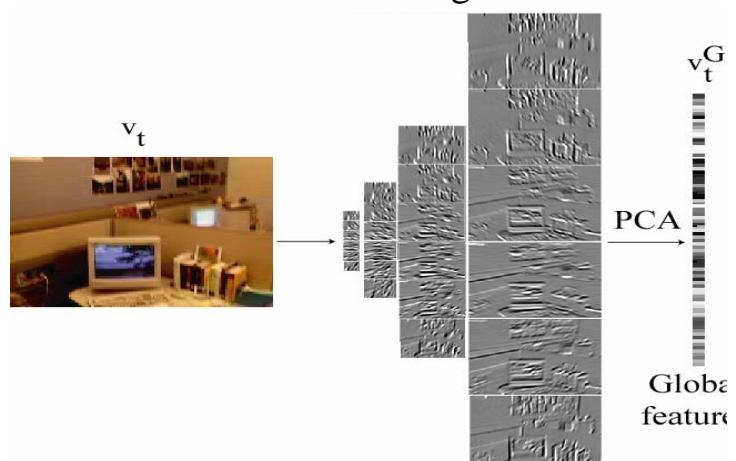
Contextual machine-vision system

- Low-dimensional representation of overall scene:
 - Gabor-filter outputs at multiple scales, orientations, locations
 - Dimensionality reduction via PCA



Feature vector for an image: the “gist” of the scene

- Compute $12 \times 30 = 360$ dim. feature vector
- Or use steerable filter bank, 6 orientations, 4 scales, averaged over 4×4 regions = 384 dim. feature vector
- Reduce to ~ 80 dimensions using PCA



Low-dimensional representation for image context

Images
80-dimensional
representation



Hardware set-up

- Wearable system
 - Gives immediate feedback to the user
 - Must handle general camera view
- Computer: Sony laptop
 - Capable of wireless link for audience display
- Designed for utility, not fashion...

Our mobile rig, version 1



Kevin Murphy

(Courtesy of Kevin Murphy. Used with permission.)

Our mobile rig, version 2.

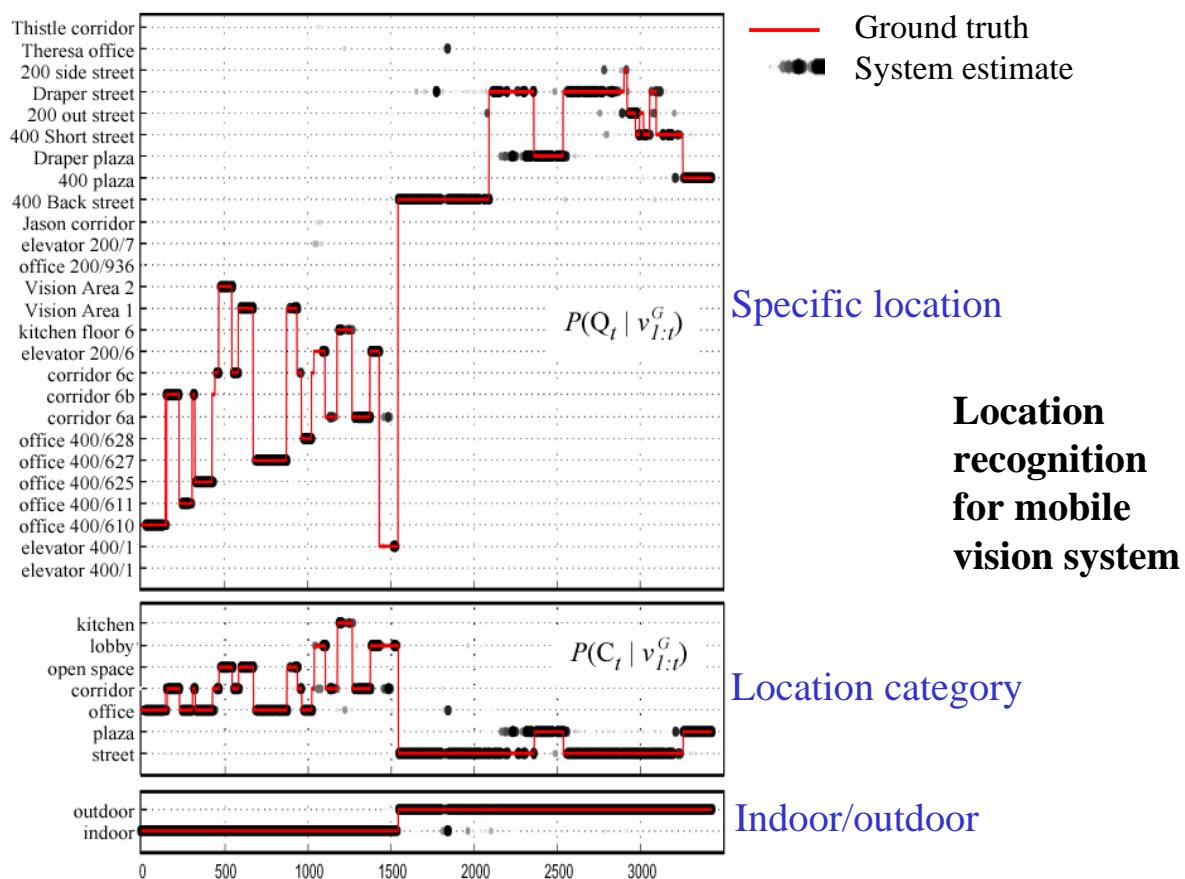


Antonio Torralba

(Courtesy of Antonio Torralba. Used with permission.)

Experiments

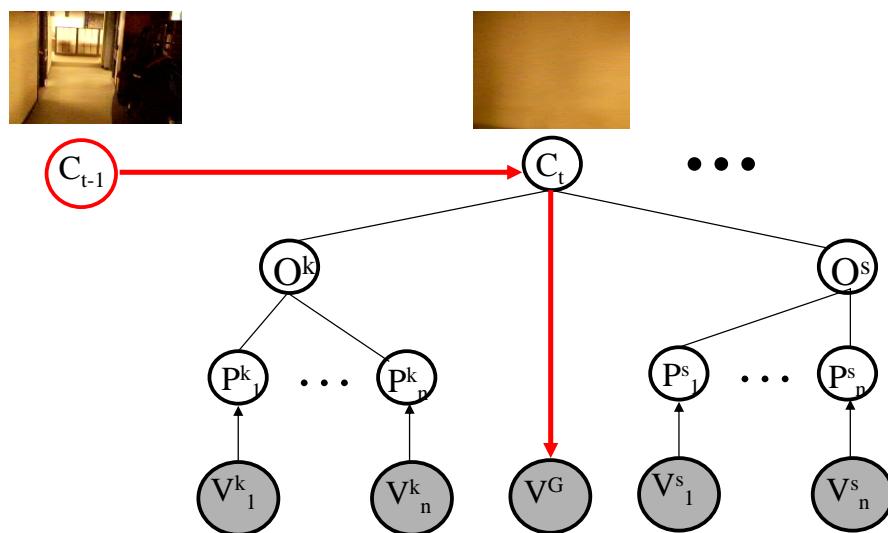
- Train:
 - Rooms and halls on 9th floor of 200 Tech. Square
 - Outdoors
- Test:
 - Interior of 200 Tech. Square, 9th floor (seen in training)
 - Interior of 400 Tech. Square (unseen)
 - Outdoors (unseen places)
- Goals:
 - Identify previously seen locations
 - Identify category of previously unseen locations



Classifying isolated scenes can be hard



Scene recognition over time

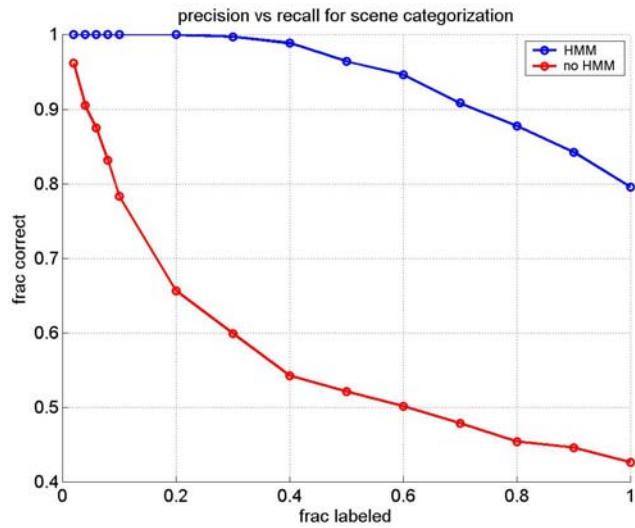


$P(C_t|C_{t-1})$ is a transition matrix, $P(v^G|C)$ is a mixture of Gaussians

Cf. topological localization in robotics

Torralba, Murphy, Freeman, Rubin, ICCV 2003

Benefit of using temporal integration



Place recognition demo

Instantaneous detection

$$\bigcirc p(q_t | v_t)$$

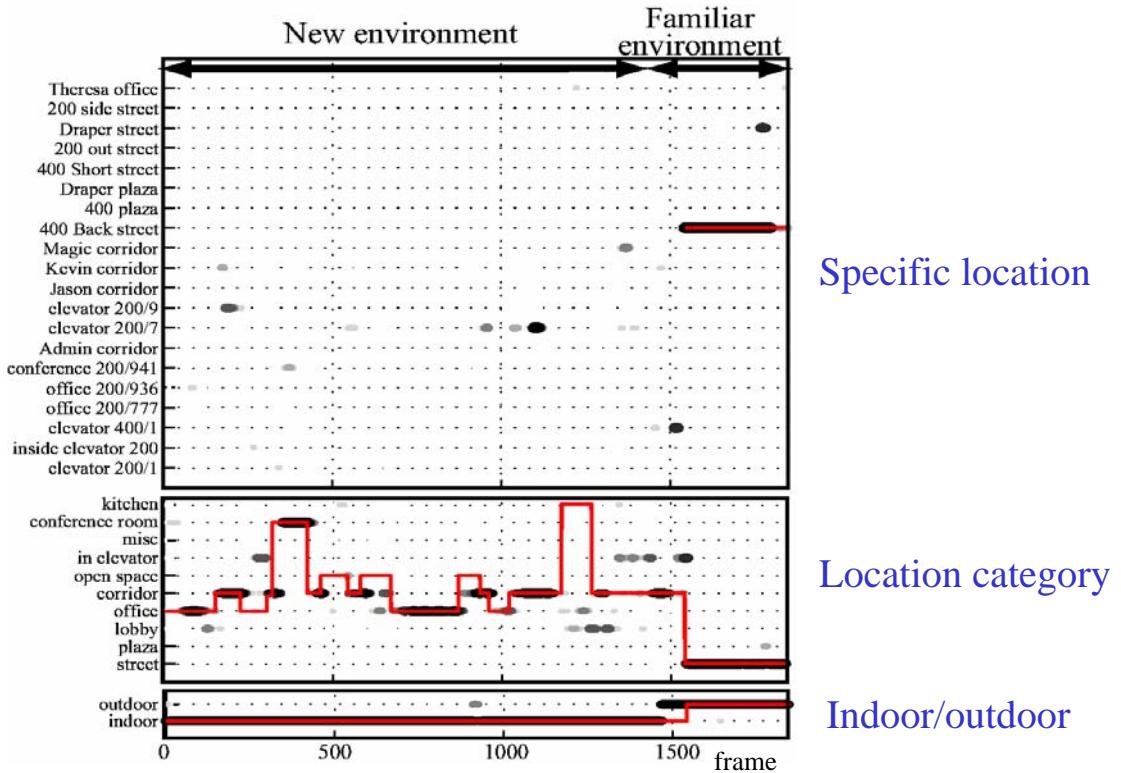
Using HMM over time

$$\bigcirc^G P(q_t | v_{1:t}^G)$$

$t=1200$ (LAB: 901)



Categorization of new places



Top-down information for object detection

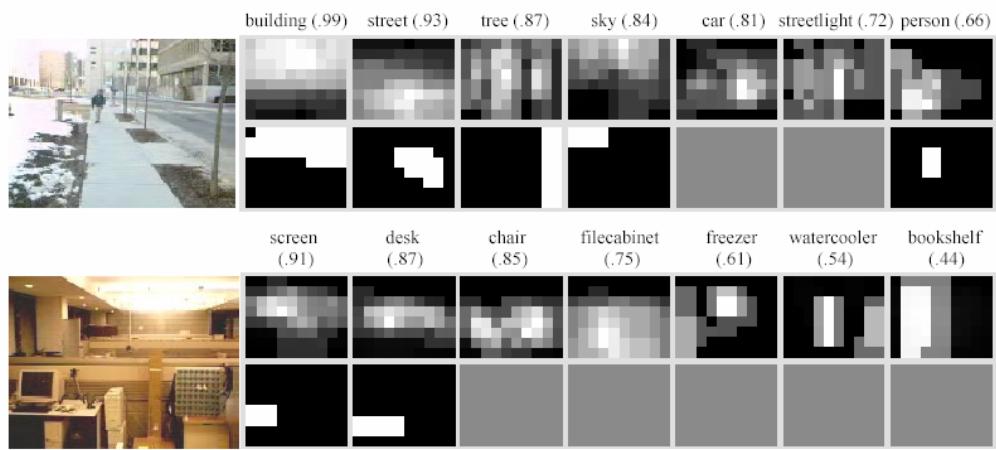


Figure 12: Some results of object localization. The gray-level images represent the probability of the objects being present at that location; the black-and-white images represent the ground truth segmentation (gray indicates absent object). Images are ordered according to $P(O_{s,i}|v_{1:t}^G)$.

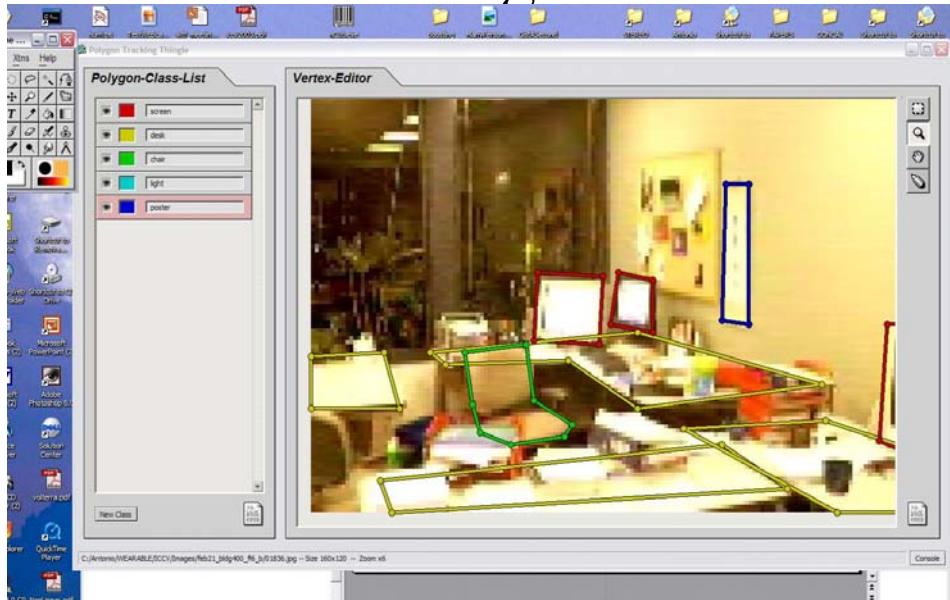
Talk outline

- Context-based vision
- Feature-based object detection
- Graphical model to combine both sources

Bottom-up object recognition

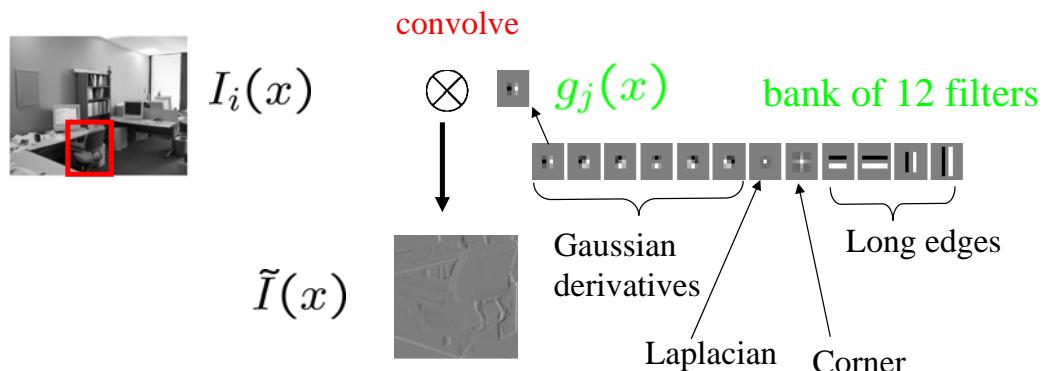
- Use labelled training set
- Use local features to categorize each object
(each view of an object)

Training data

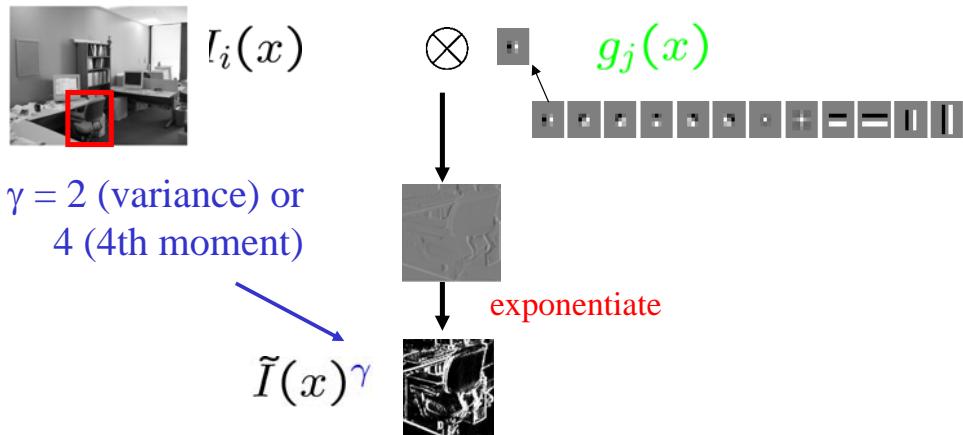


- Hand-annotated 1200 frames of video from a wearable webcam
- Trained detectors for 9 types of objects: bookshelf, desk, screen (frontal), steps, building facade, etc.
- 100-200 positive patches, > 10,000 negative patches

Feature vector for a patch: step 1



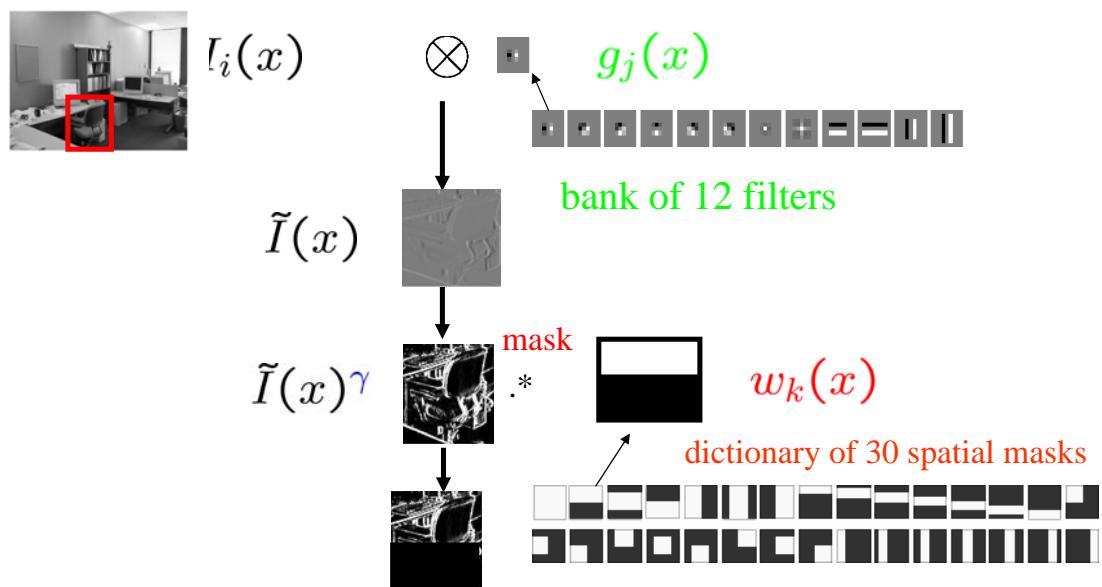
Feature vector for a patch: step 2



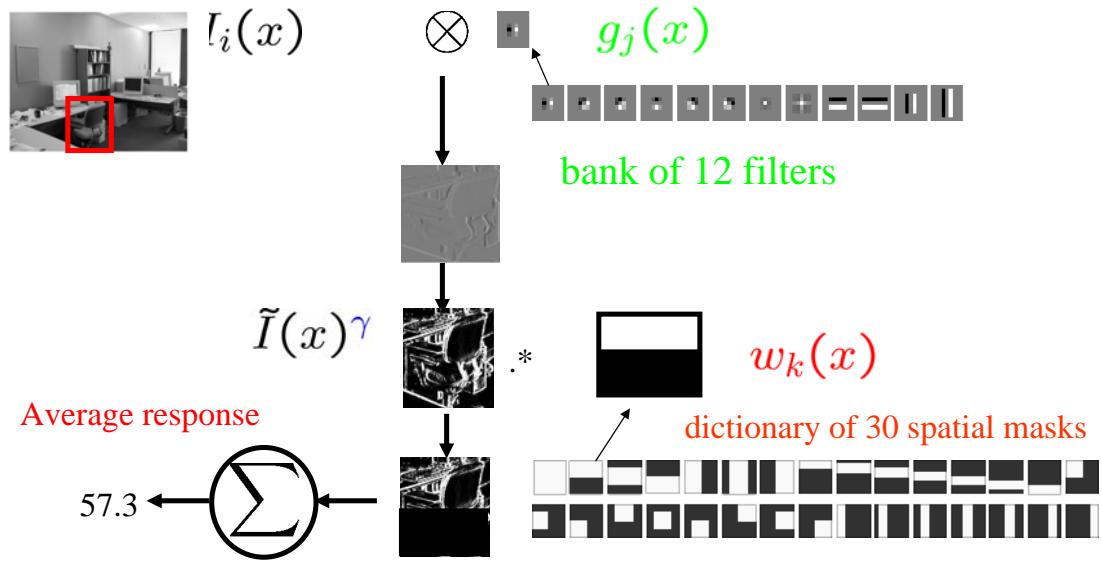
Kurtosis $K = \frac{EX^4}{[EX^2]^2}$ characterizes shape of filter response

Useful for texture analysis

Feature vector for a patch: step 3



Feature vector for a patch: step 4



Summary: Features

$$f_i(k) = \sum_x w_k(x) (|I(x) \otimes g_k(x)|^{\gamma_k})_i$$

Diagram illustrating the formula for the k'th feature of i'th patch:

- The formula is $f_i(k) = \sum_x w_k(x) (|I(x) \otimes g_k(x)|^{\gamma_k})_i$.
- The image $I(x)$ is processed by a bank of 12 filters $g_k(x)$ (represented by a grid of 12 small images).
- The result is a feature map $|I(x) \otimes g_k(x)|^{\gamma_k}$.
- The feature map is multiplied by a dictionary of 30 spatial masks $w_k(x)$ (represented by a grid of 30 small images).
- The final output is the k'th feature of i'th patch.

12 x 30 x 2 = 720 features. Special cases include:

- g_k = delta function, w_k = Haar wavelets – Viola & Jones, Poggio et al
- $f_i(\gamma)=4/ f_i(\gamma=2)$ gives kurtosis for texture analysis
- w_k mask to capture spatial arrangement of parts
- Rectangular masks support integral image trick for fast computation

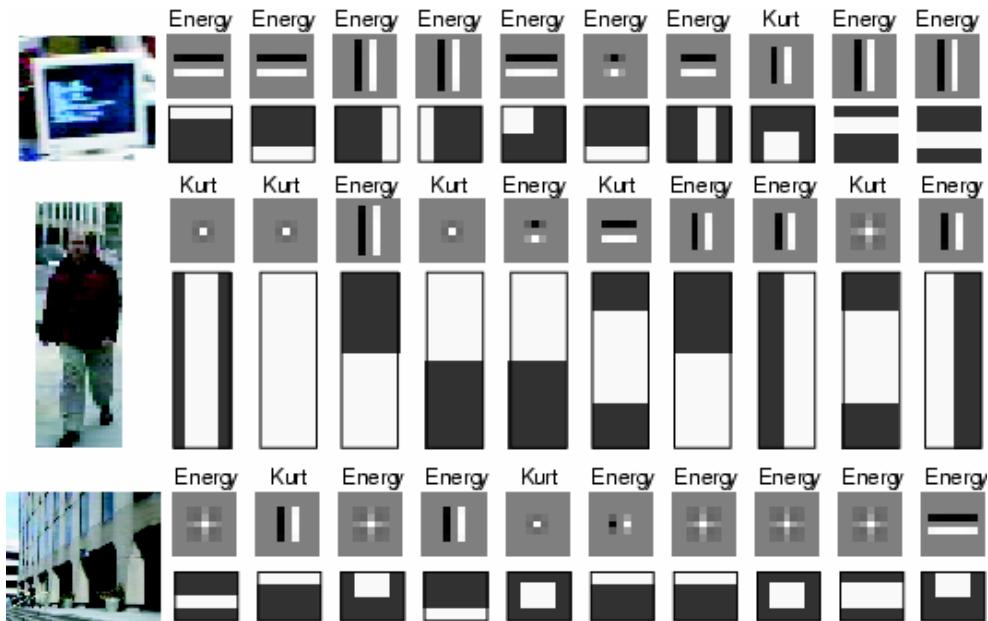
Classifier: boosted features

- Output is $b = \sum_t \alpha_t h_t(\vec{f})$ where
 - f = feature vector for patch
 - $h_t(f)$ = output of weak classifier at round t
 - α_t = weight assigned by boosting
- Weak learners are single features:
 $h_t(f)$ picks best feature and threshold:
$$h_t(\vec{f}) = (f(j, k, \gamma) > \theta)$$
- ~500 rounds of boosting
- ~200 positive patches, ~ 10,000 negative patches
- No cascade (yet)

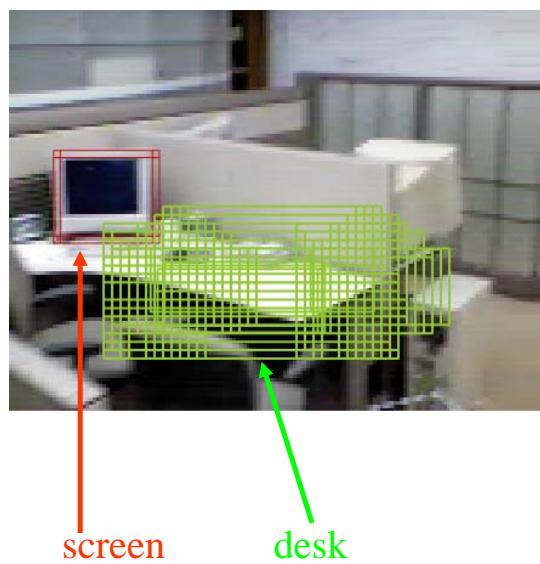
Viola & Jones, IJCV 2001

Boosting demo

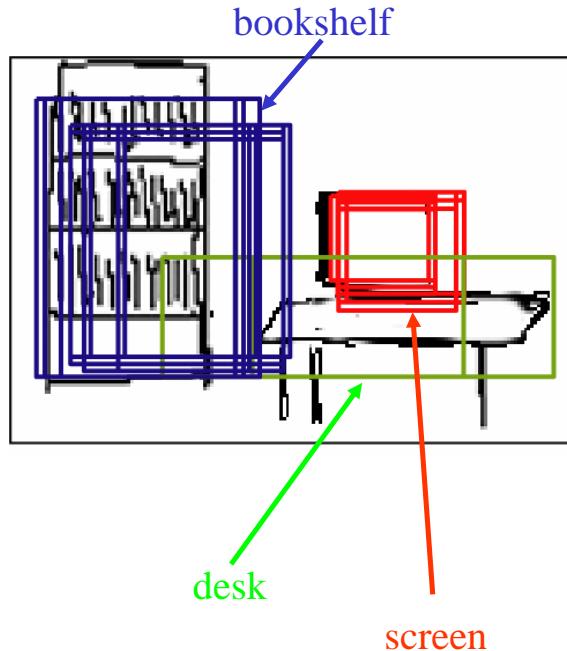
Examples of learned features



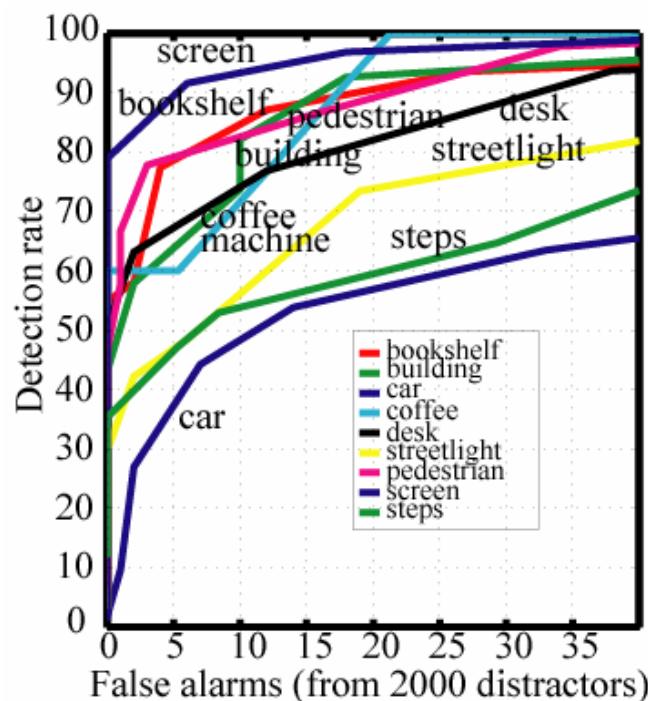
Example detections



Example detections



Bottom-up detection: ROC curves

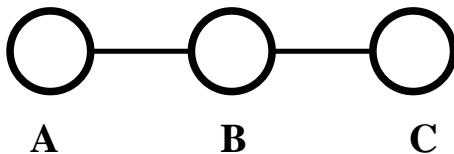


Talk outline

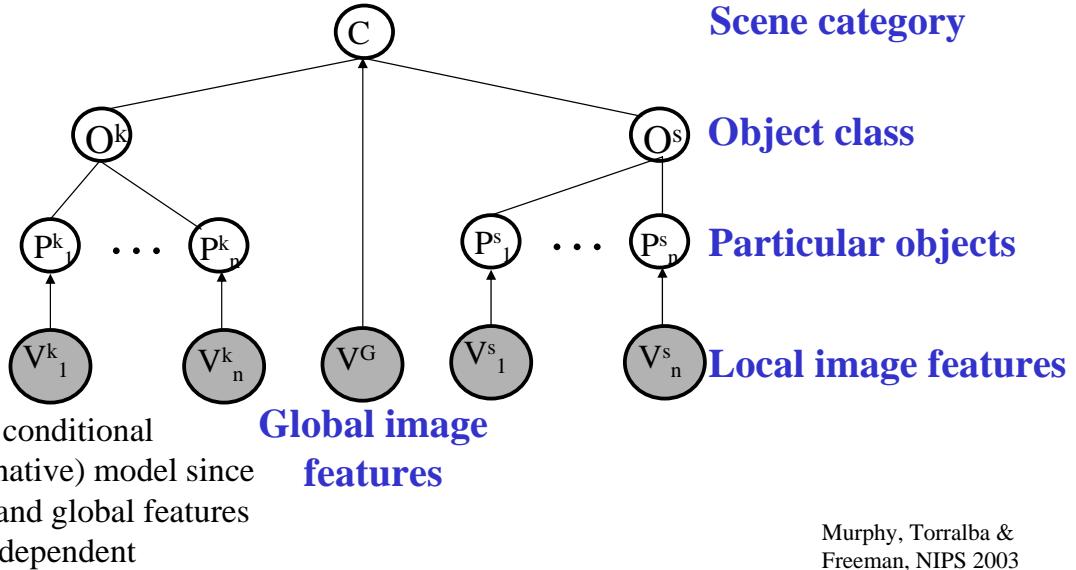
- Context-based vision
- Feature-based object detection
- Graphical model to combine both sources

Probabilistic models: graphical models

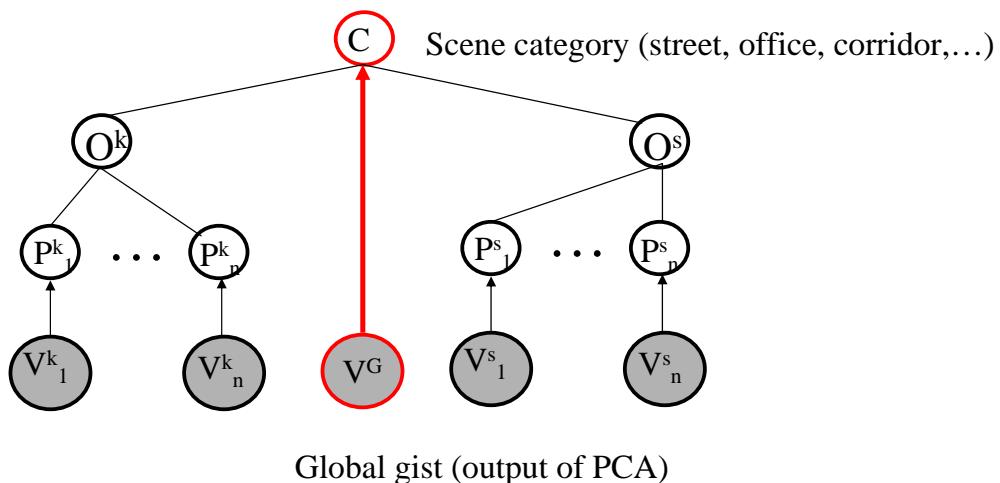
- Tinker toys for probabilistic models
- Build up complex models from simple components describing conditional independence assumptions.
- Standard inference algorithms let you combine evidence from different parts of the model.



Combining global scene information with local detectors using a probabilistic graphical model

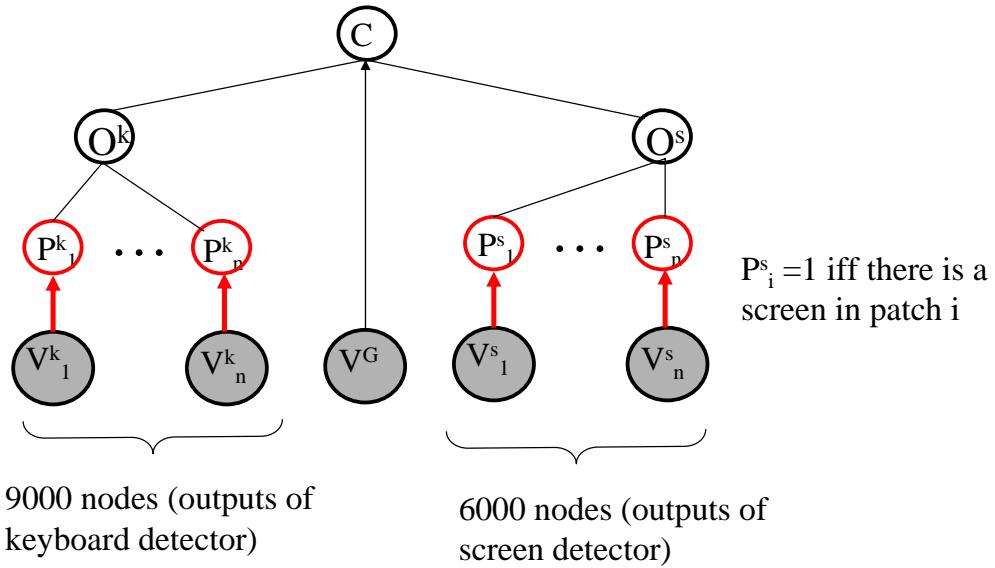


Scene categorization using the gist



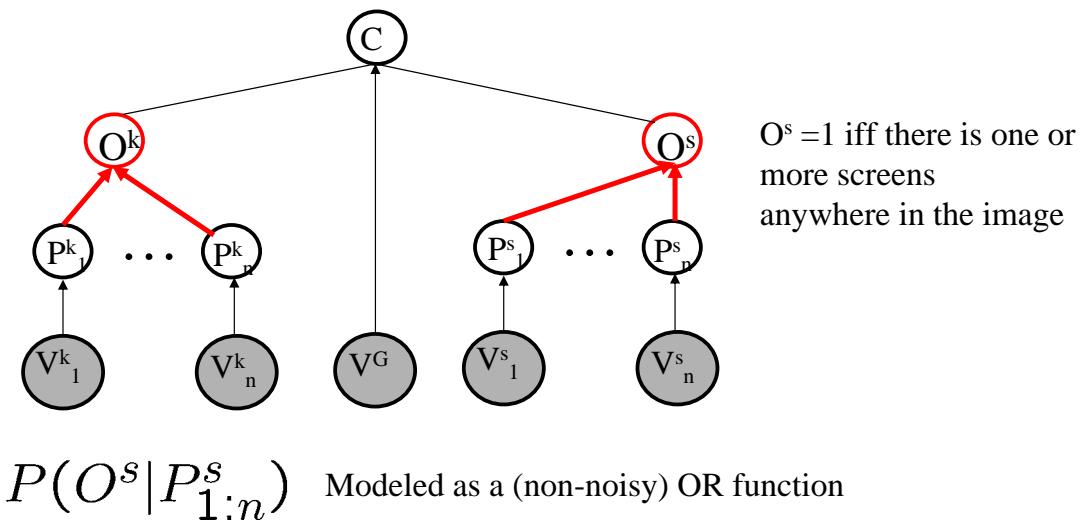
$P(C|v^G)$ modeled using multi-class boosting or by a mixture of Gaussians

Local patches for object detection and localization



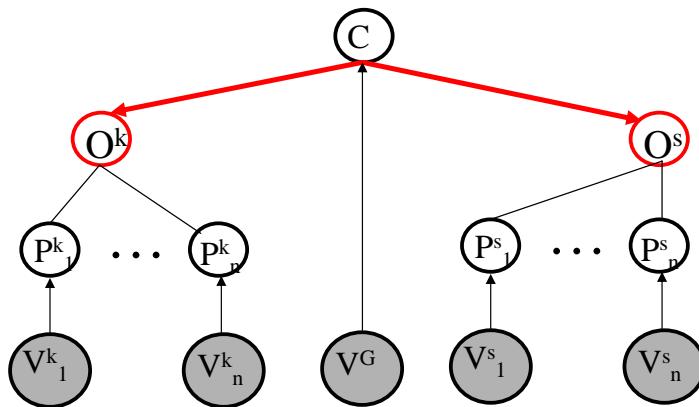
$$P(P_i^s = 1 | v_i^s) = \sigma(\lambda^T [1; b(v_i^s)])$$

Location invariant object detection



O nodes useful for image retrieval, scene categorization and object priming

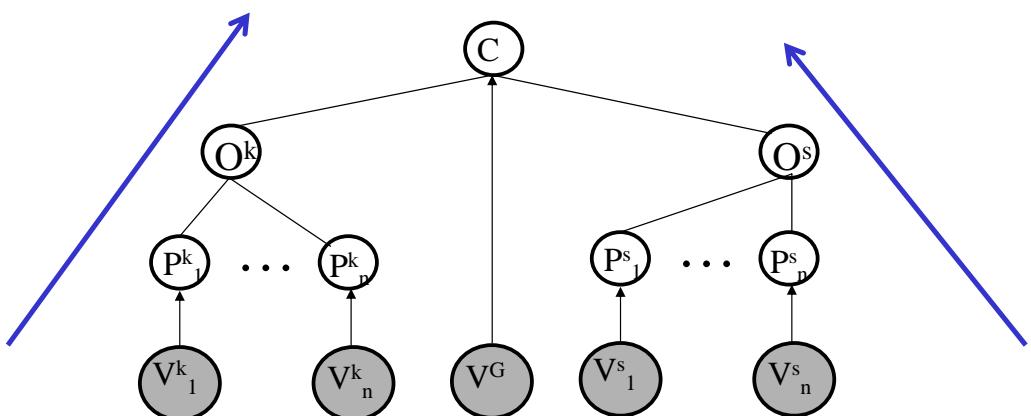
Probability of object given scene



$$P(O^s | C) \quad \text{estimated from co-occurrence counts}$$

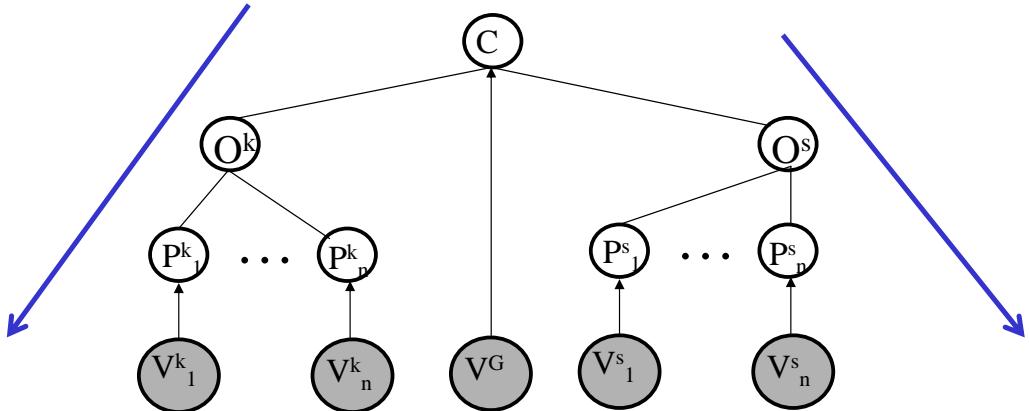
Inference in the model

Bottom-up, from leaves to root



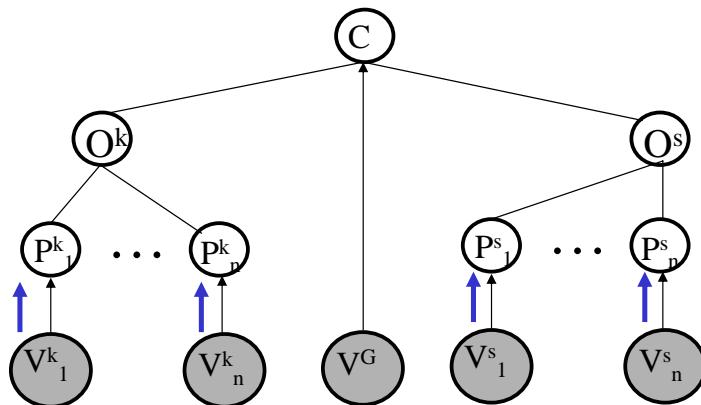
Inference in the model

Top-down, from root to leaves



1. Run detectors and classify patches

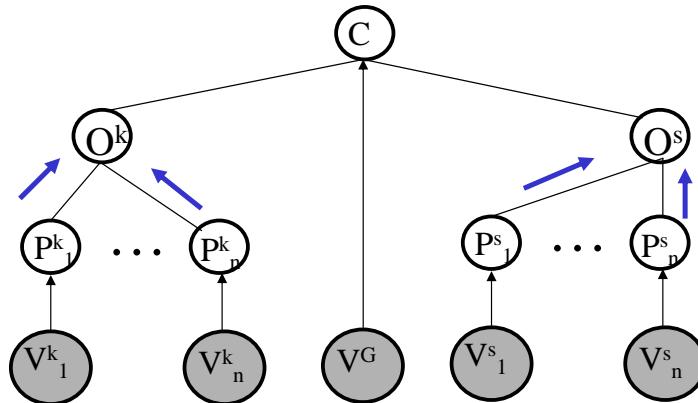
“Boy, exhaustive search is tiring!”



$$P(P_i^s = 1 | v_i^s)$$

2. Infer object presence given detectors

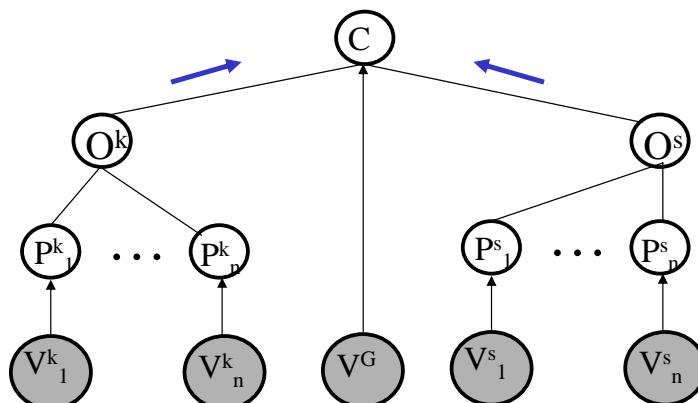
“Some screen detectors fired, so there’s probably a screen somewhere”



$$P(O^s = 1 | v_{1:n}^s)$$

3. Classify scene using parts (objects)

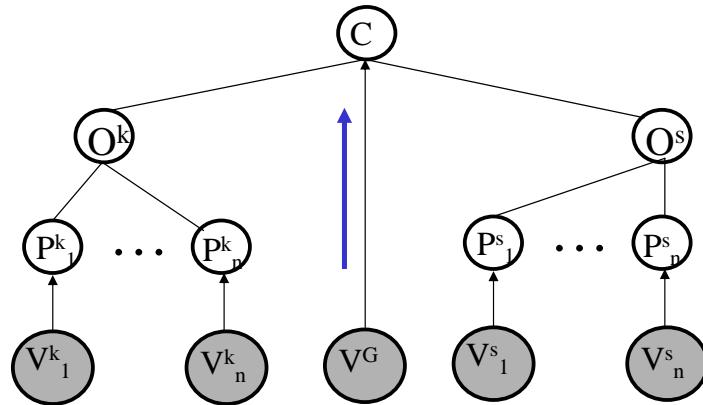
“I think I saw a screen and a car, so I may be in an office or a street”



$$P(C | v_{1:n}^s, v_{1:n}^k)$$

4. Classify scene holistically (gist)

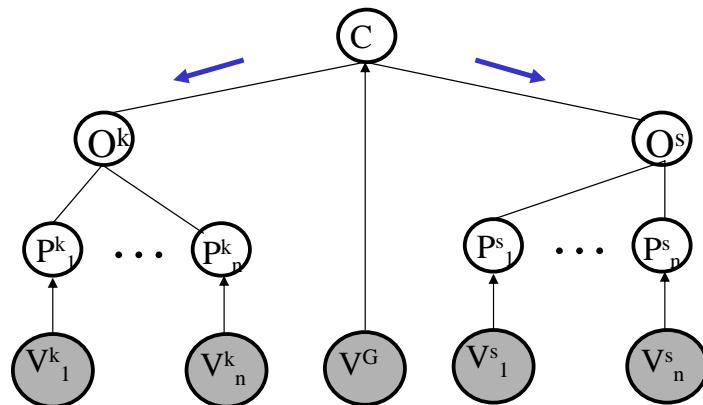
“This looks like a street to me”



$$P(C|v^G, v_{1:n}^s, v_{1:n}^k) \propto P(v^G|C)P(C|v_{1:n}^s, v_{1:n}^k)$$

5. Update object estimates using scene

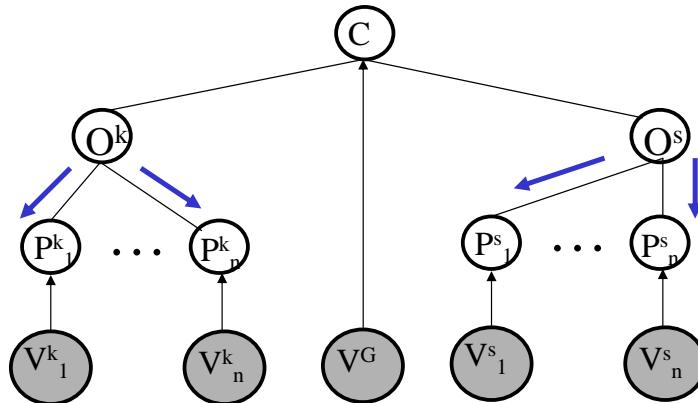
“Since I’ve decided I’m in a street, there is no screen in the image”



$$P(O^s = 1|v^G, v_{1:n}^s, v_{1:n}^k)$$

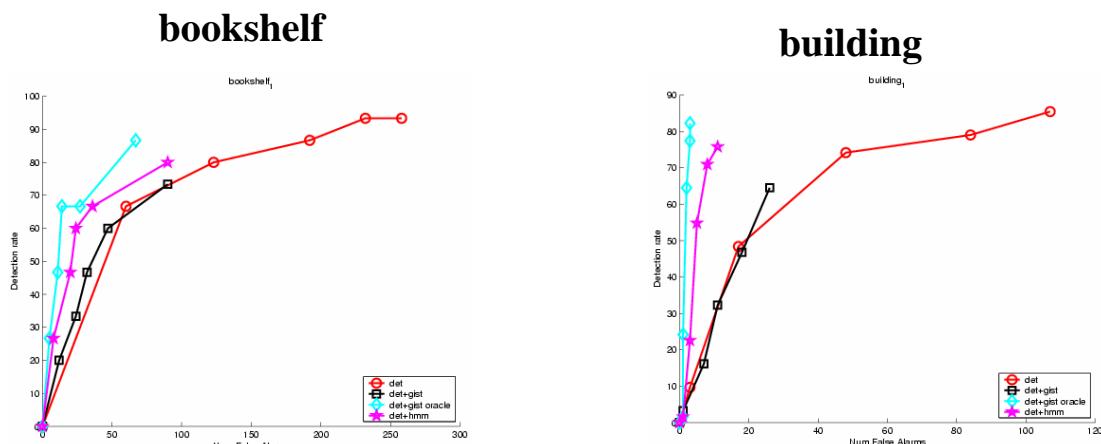
6. Update patch estimates using objects

“Since there’s no screen in the image, this patch is a false positive”



$$P(P_i^s = 1 | v^G, v_{1:n}^s, v_{1:n}^k)$$

Effect of context on object detection

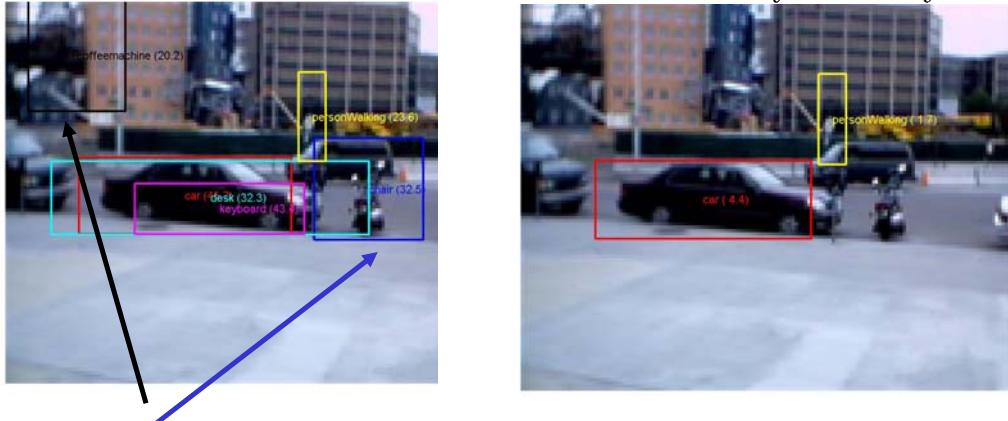


- **detector**
- **detector + v_t^G**
- **detector + $v_{1:t}^G$ (hmm)**
- **detector + scene oracle**

Example of object priming using gist

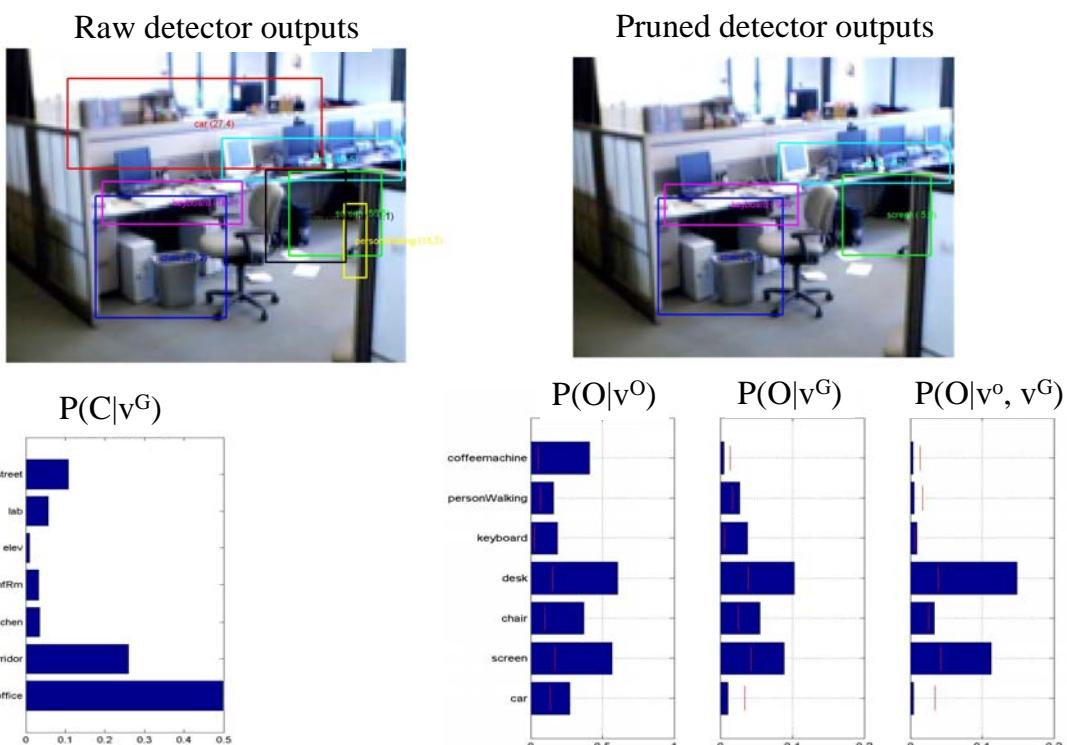
For each type of object, we plot the single most probable detection if it is above a threshold (set to give 80% detection rate)

$$\arg \max_i P(P_i^c = 1 | v_i^c) \quad \arg \max_i P(P_i^c = 1 | v_i^c, v^G)$$

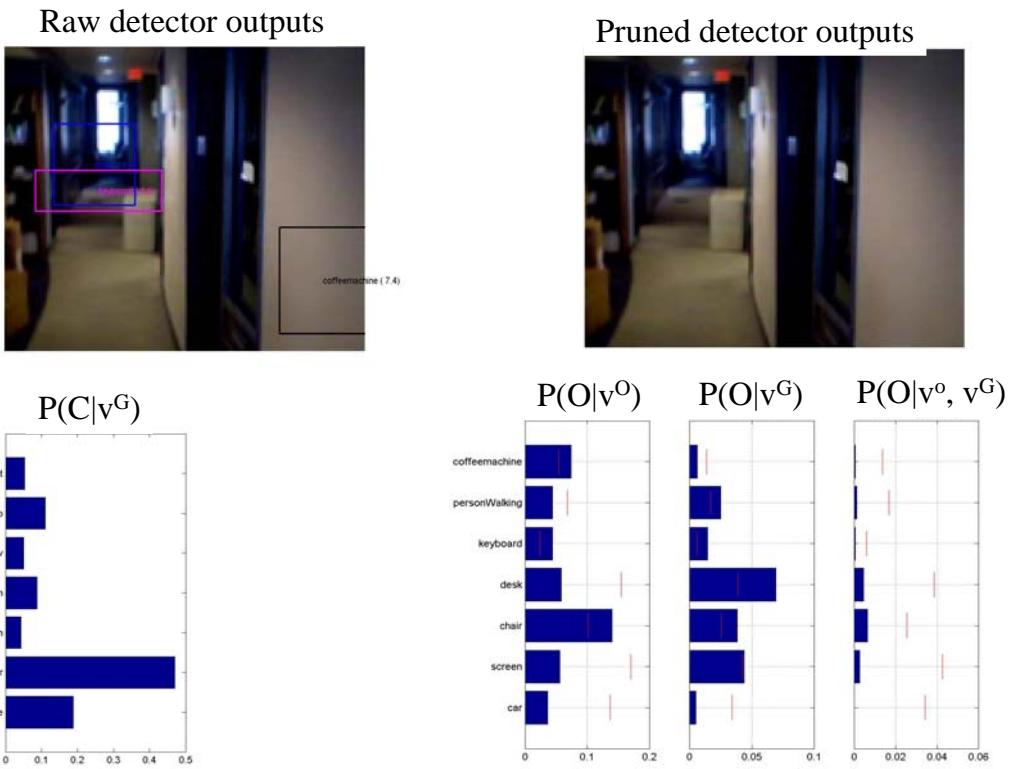


If we know we are in a street, we can prune false positives such as chair and coffee-machine (which are hard to detect, and hence must have low thresholds to get 80% hit rate)

Pruning false positives using gist



Pruning false positives using gist



Top-down and bottom-up object detection



Video input

Likely location for a car,
given current context

Detected car

Best training set wins

- Character recognition
- Speech recognition

Computer vision training set options

- Real world data, hand labeled
 - Example: Sowerby/BAE database
 - In general: expensive and slow.
- Real world, partially labeled
- Synthetic world, automatically labeled.
 - Will training there transfer to the real world?

Research goals

- Scale up: develop efficient system to recognize 1000 different objects, generalizing current feature detection cascades.
- Train exhaustively.
- Apply in wearable or other real-world systems
 - Lifelog
 - VACE

end

Future directions

- Improve local-feature-based object detection.
 - Training set
 - Efficient use of local feature information.
- Include temporal information, more than just a single HMM for the global scene context.

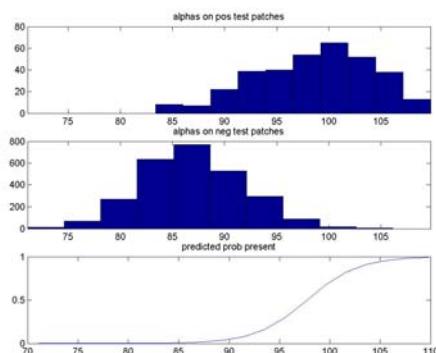
Overview of talk

- ✓ Why scene context?
 - Disambiguate local features
 - Reduce search space
- Data set
- Object detection
 - Features
 - Classifier
- Scene categorization and object priming
- Combining global scene information with local detectors using a probabilistic graphical model
- Scene categorization over time
- Location/scale priors using the scene and other objects
- Summary/ future work

Classifier: based on boosting

- Weighted output is $h = \sum_t \alpha_t h_t(f)$, where
 - f = feature vector for patch
 - $h_t(f)$ = output of weak classifier at round t
 - α_t = weight assigned by boosting
- $h_t(f)$ picks best feature f_k and corresponding threshold to minimize classification error on validation set
- 100-500 rounds of boosting

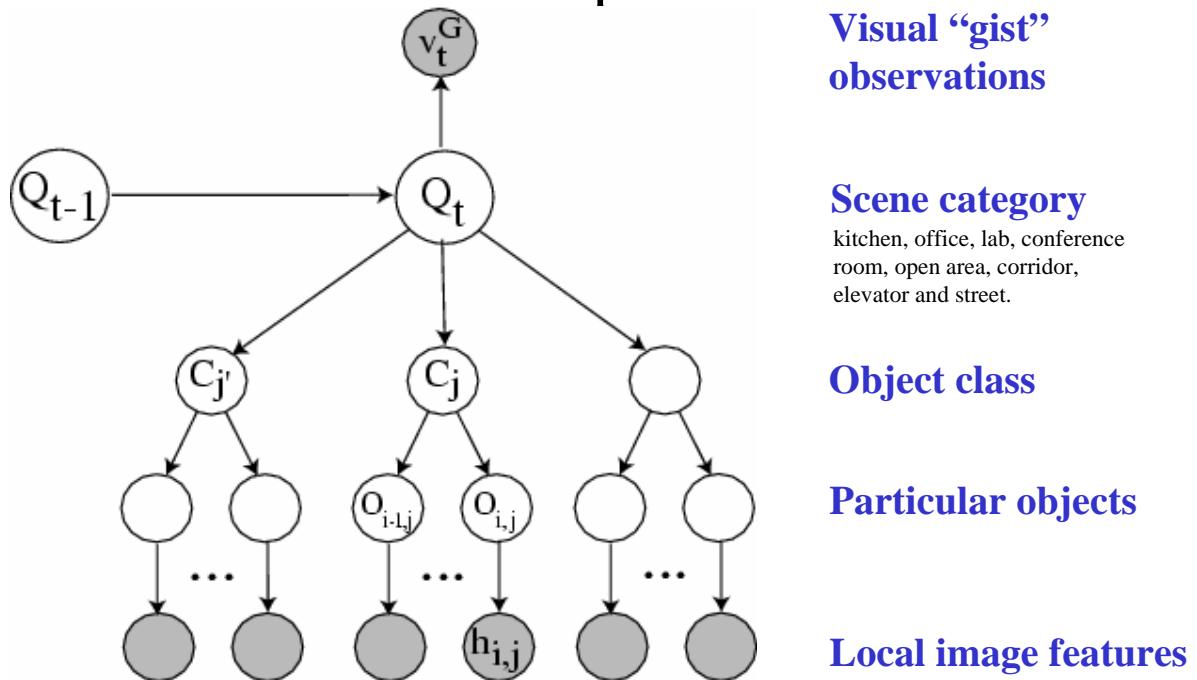
Converting boosting output to a probability distribution



$$P(P_i^k=1|b) = \sigma(\lambda^T [1; b])$$

sigmoid weights Offset/bias term

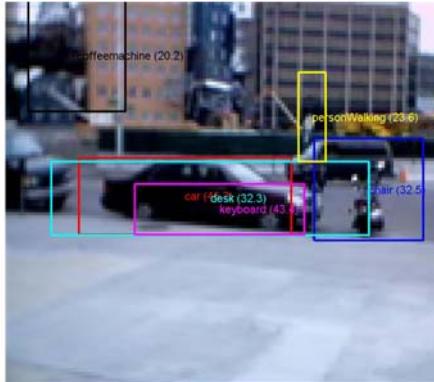
Combining top-down with bottom-up: graphical model showing assumed statistical relationships between variables



Pruning false positives using gist

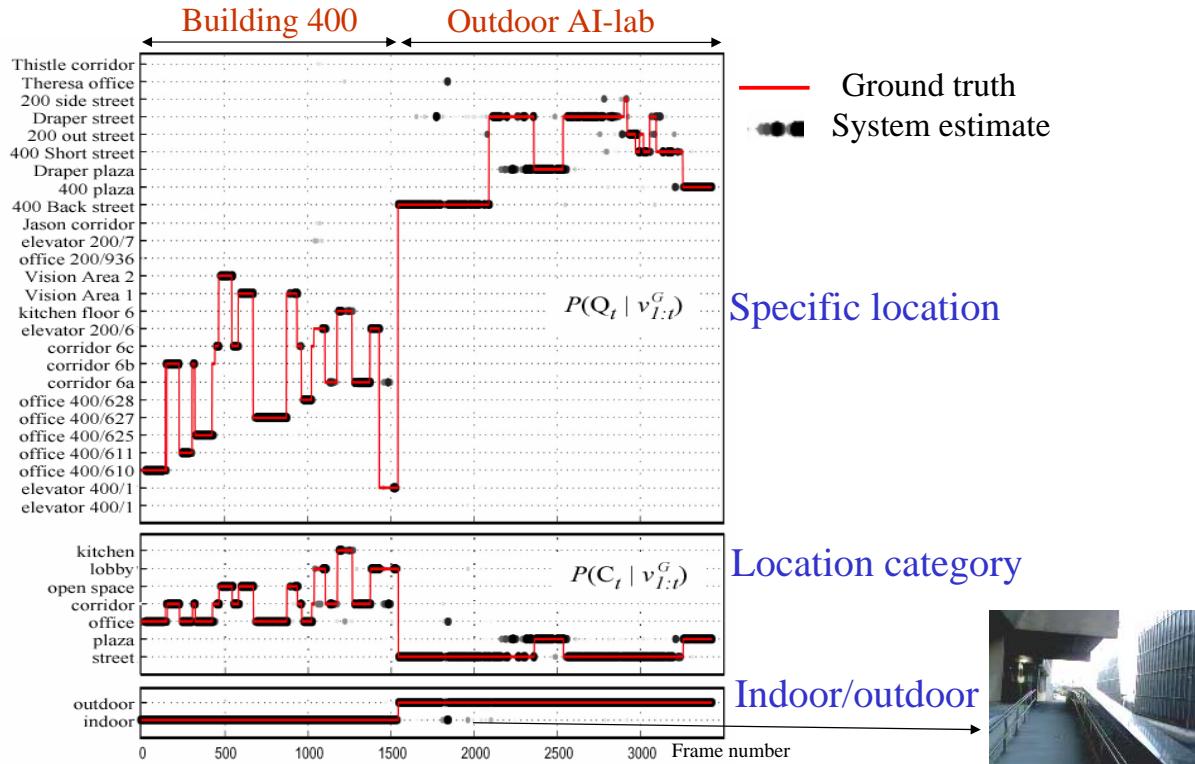
For each type of object, we plot the single most probable detection if it is above a threshold (set to give 80% detection rate)

$$\arg \max_i P(P_i^c = 1 | v_i^c) \quad \arg \max_i P(P_i^c = 1 | v_i^c, v^G)$$

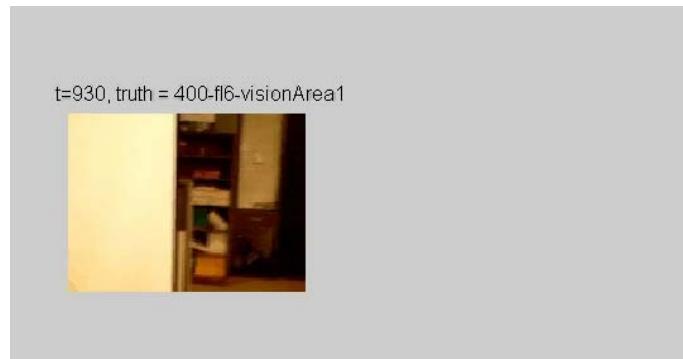


Using the gist, we figured out we’re in a street, so probability of chair and coffee machine drops below threshold.

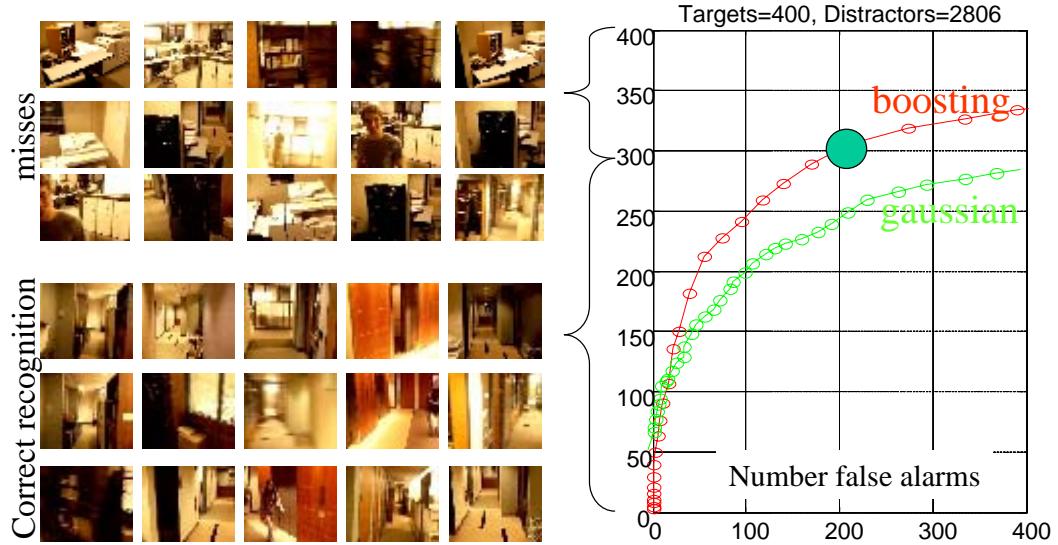
Place and scene recognition using gist



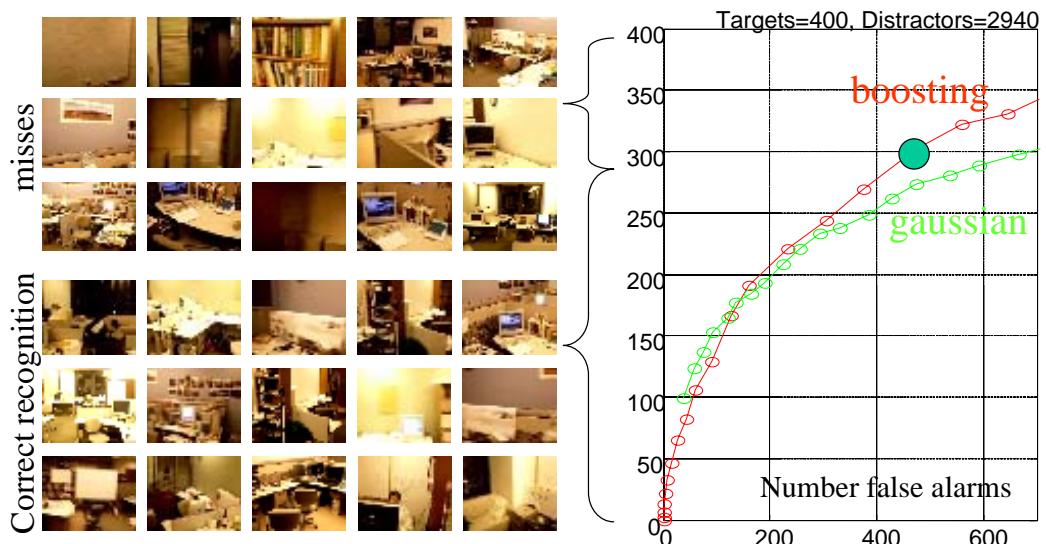
Place recognition demo



Corridor recognition



Office recognition



Scene categorization using the gist

Estimate $P(C|v^G)$ using multi-class boosting or mixture of Gaussians for 7 pre-chosen categories.



Object priming using scene category

Compute $P(O|v^G) = \sum_{c=1}^7 P(O|c)P(c|v^G)$ for 9 object classes O

where $P(O|C)$ is estimated by counting co-occurrences in labeled images

